LearnDelphi.org

Coding Boot Camp

learndelphi.org/boot-camp-2022

LearnDelphi.org

Session : Core14

Introduction to Object Oriented Programming

About Bertie Buitendag

Tshwane University of Technology
We empower people

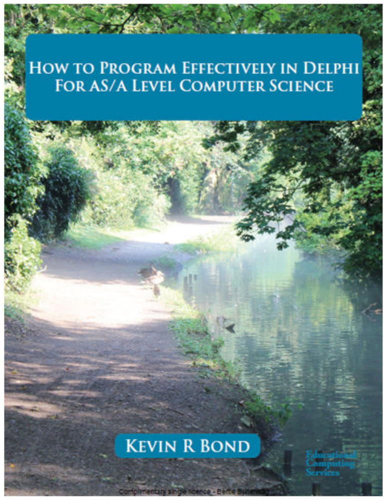
LearnDelphi.org
Coding Boot Camp 2022

- Lecturer at the department of Computer Science at the Tshwane University of Technology in South Africa
- Passionate about Delphi since Borland Delphi Version 2.0 (1996)
- Assisted with the development of the IT secondary school curriculum in SA
- Embarcadero MVP and Delphi evangelist
- Lives in Pretoria in sunny SA 🇿🇦
- Love comedy and are in the aquarium fishing hobby

Session outline

- Basic introduction to the core concepts of OOP in Delphi
- Step by step example
- Example 2
- Q & A

Textbook reference chapters


How to Program Effectively in Delphi
For AS/A Level Computer Science
Kevin R Bond
Educational Computing Services

Chapters 18 to 20

Core concepts - OOP

Object Oriented Programming (OOP) - Definitions.

OOP is a programming methodology which incorporates all the traditional features of procedural programs but with several enhancements. **OOP however requires a different way of thinking** and adds several new concepts to programming. (Farrel, 2002:10)

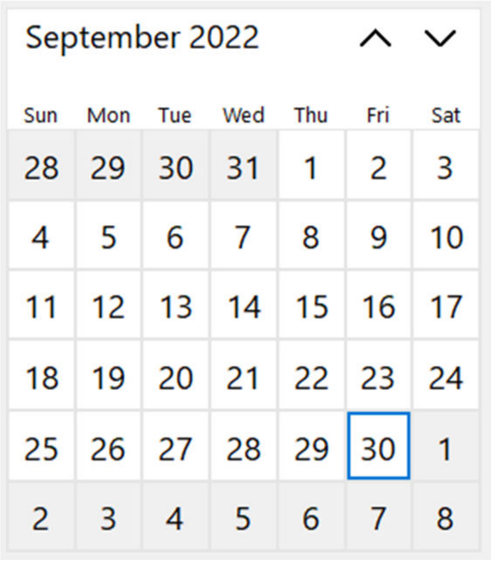
OOP allows programmers to reuse and modify existing objects resulting in faster program creation. OOP languages are **often event driven**, where events are actions to which a program responds, such as pressing a key on a keyboard, typing a value in a text box etc. (Shelly, et al. 2005, I.17)

Object Orientation

OO or Object Orientation tries to modal real word objects (in a software format) in terms of its actions and its attributes.

In other words, OOP aims to create software that models or simulate real world objects. OOP aims to simulate the object in terms of its attributes (the characteristics of the object) and its behaviour (what the object can do).

CalenderPicker



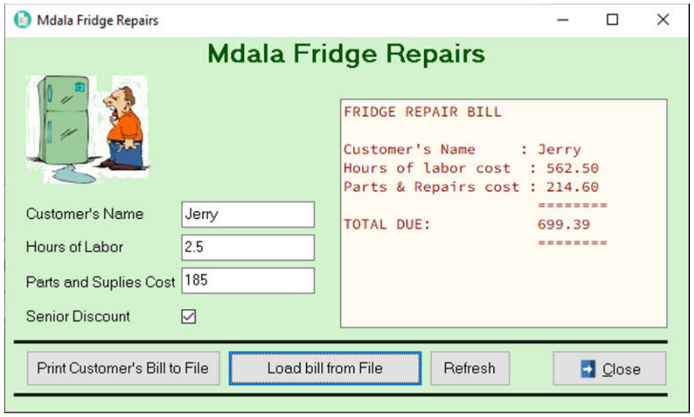
September 2022

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Object Orientation

OO or Object Orientation tries to modal real word objects (in a software format) in terms of its actions and its attributes.

Many programming languages which support EDP (Event Driven Programming) are based on OO. Users create computer programs which models real world scenarios. The program interface, normally a GUI is composed of various components which reacts to user events such as clicking on a button opening a form etc.



Customer's Name: Jerry
Hours of Labor: 2.5
Parts and Suplies Cost: 185
Senior Discount ☑

```
FRIDGE REPAIR BILL

Customer's Name      : Jerry
Hours of labor cost  : 562.50
Parts & Repairs cost : 214.60
                       ========
TOTAL DUE:             699.39
                       ========
```

Procedural vs OO paradigm

Procedural

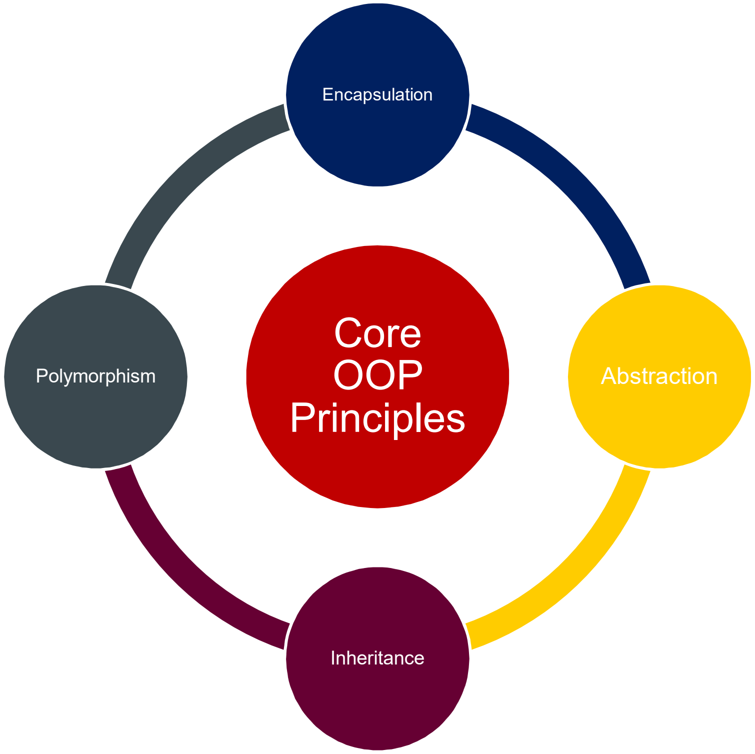
- Implements Type-definitions e.g. (Record Structures)
- Functions & Procedures
- Variables
- Procedure cannot be connected to a record
- No Inheritance
- No real information hiding.
- No data encapsulation.
- Error handling, programmers' responsibility

Object Oriented

- Implements object members.
- Methods -> Procedures & Functions.
- Attributes / Members
- Allows Inheritance
- Restrictions to object change may be implemented.
- Error handling can be implemented.
- Allows true encapsulation.
- Information hiding

Core concepts and terms

LearnDelphi.org
Coding Boot Camp 2022



In order for the programming language to be considered a true OO programming language it must support the following OO features.

Abstraction and Classes

Abstraction

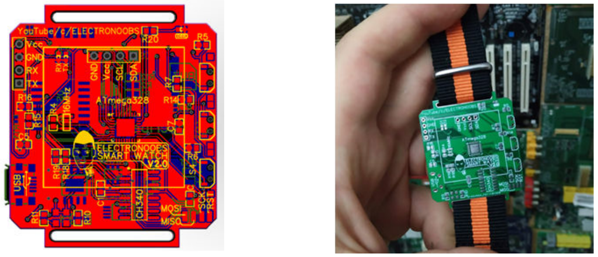
Class

Abstraction

Abstraction is the ability to engage with a concept while ignoring some of its details. (Bond, 2021)

With abstraction only the required details are shown, and all unnecessary details are hidden.

FocusFit Pro-T500 Smartwatch and Fitness Tracker



Source: https://electronoobs.com/eng_arduino_tut61.php

Abstraction vs Abstract

Two basic definitions of the two terms:

Abstraction

- Is the process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristics
  (https://www.techtarget.com/whatis/definition/abstraction)

Abstract

- Difficult to understand
- Thought of apart from any particular instances or material objects; not concrete.
  (https://www.yourdictionary.com/abstract)

Abstraction conceptualised

Think about a datatype for each of the following variable names:

Surname	→	String	
Height	→	Real	
PageCount	→	Integer	
TeacherSalary	→	Real / Currency	(Byte ☺)
GameRectangle	→	???	
Product	→	???	

Abstraction conceptualised

GameRectangle

The GameRectangle abstraction can be defined using descriptors for its attributes, i.e.,

- Height → Real
- Width → Real
- Colour → String

We can describe and define abstract concepts using concrete datatypes.

Abstract Data Type

ADT (Abstract Data type) can therefore be seen as: **a user-defined data type** consisting of (being described by) several simple data types.

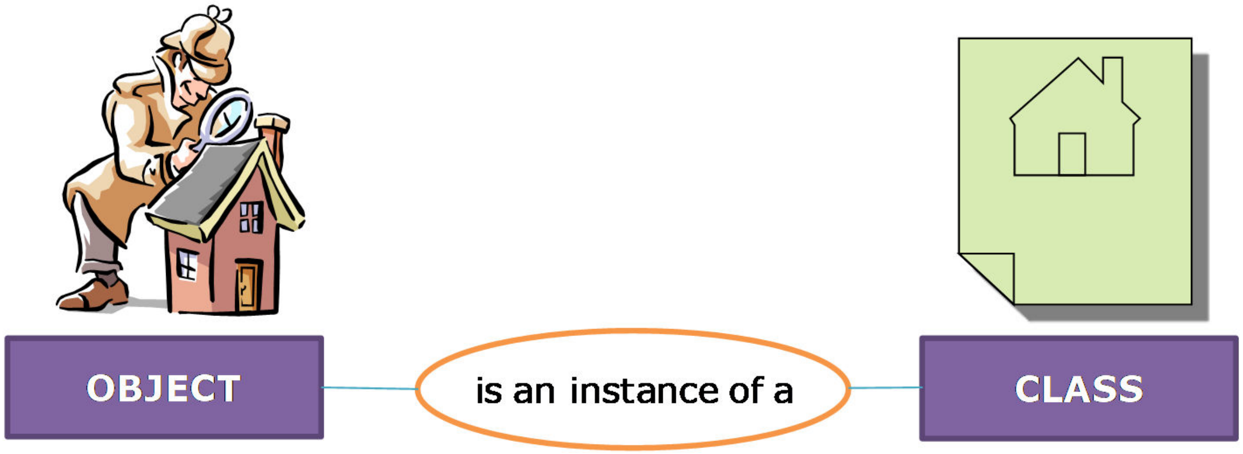
E.g. Brick (length, width, height)
Date (year, month, day)
Time (hours, minutes, seconds)

An ADT is used to define objects that cannot be defined by using concrete / simple data types.

Abstraction and Classes

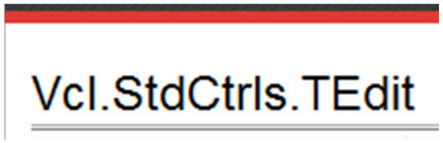
Classes

A class is a template from which actual objects may be created. The class definition lists the characteristics of the object that are to be recorded. (Bond, 2021).

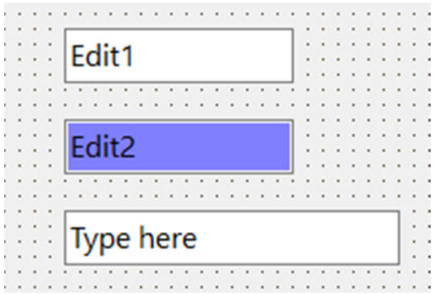


VCL (Visual component library)

VCL is an acronym for the Visual Component Library, a set of visual components for rapid development of Windows applications in the Delphi language. VCL contains a wide variety of visual, non-visual, and utility **classes** for tasks such as Windows application building, web applications, database applications, and console applications.



Vcl.StdCtrls.TEdit



Edit1

Edit2

Type here

Abstraction and Classes

Class

Object

Class

- A class can be used to define an ADT
- Defines what an object looks like in terms of its attributes and its behaviour

Object

- It is a ready-to-use representation of a class

A class in object-oriented programming is also an abstraction of the generalisation type (Bond, 2021).

Abstraction and Classes

Encap-
sulation

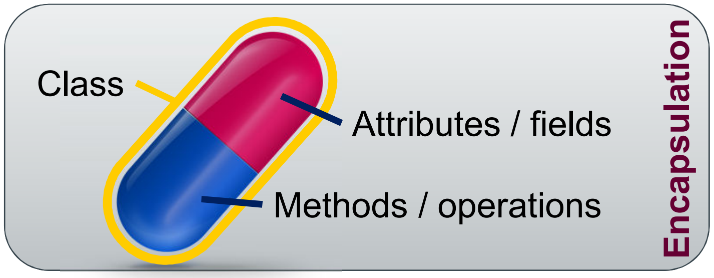
Class

Encapsulation

- Is a way of hiding the precise implementation details of a class and binds (groups together and protects) the data (i.e., attributes) and the methods of the class.
- The goal of encapsulation is to create objects that can be re-used whenever they are appropriate. This re-use should occur without you having to be concerned about special requirements, nor about having to manage undesirable side effects. Defines what an object looks like in terms of its attributes and its behaviour. (CodeGear 2007)

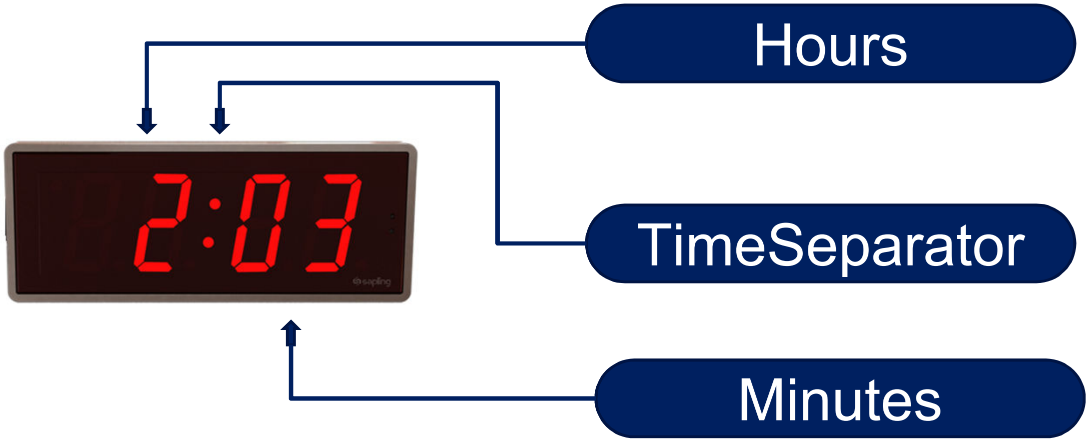
Object

- It is a ready-to-use representation of a class



Step by Step Example

TMyTime

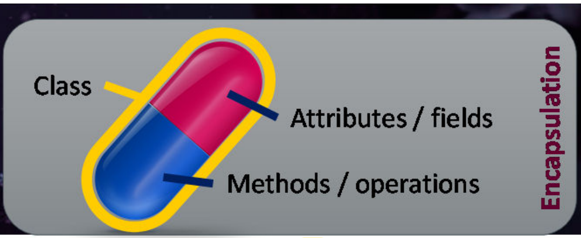


Hours

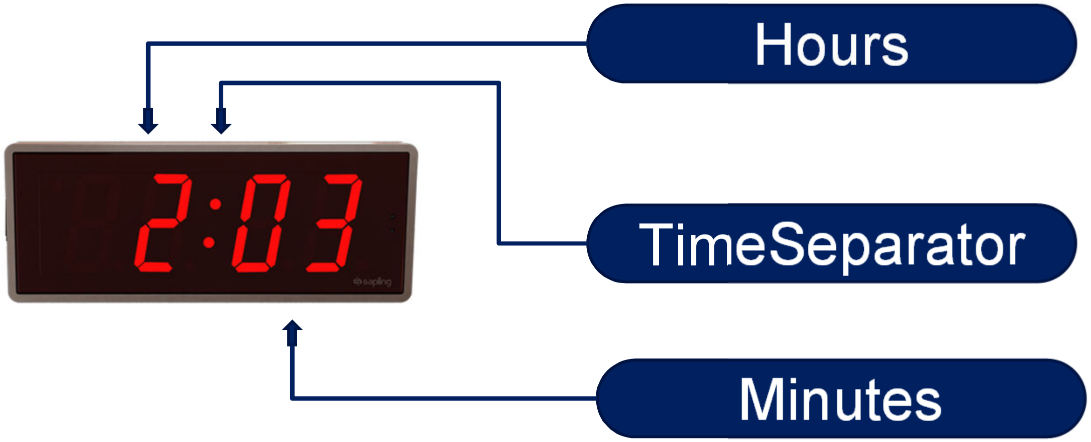
TimeSeparator

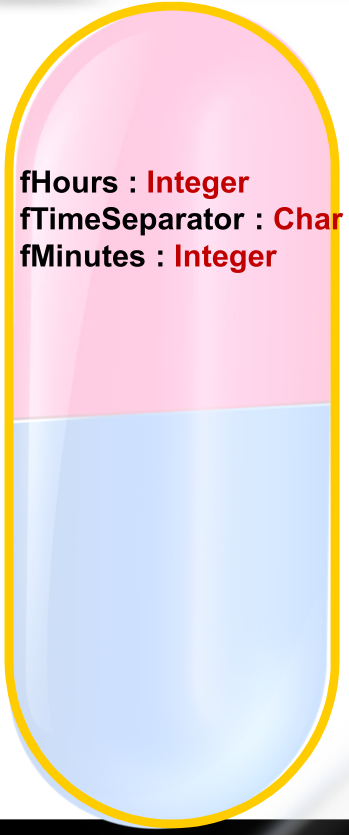
Minutes

Step by Step Example



TMyTime

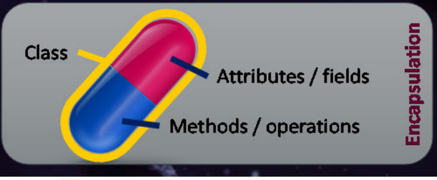




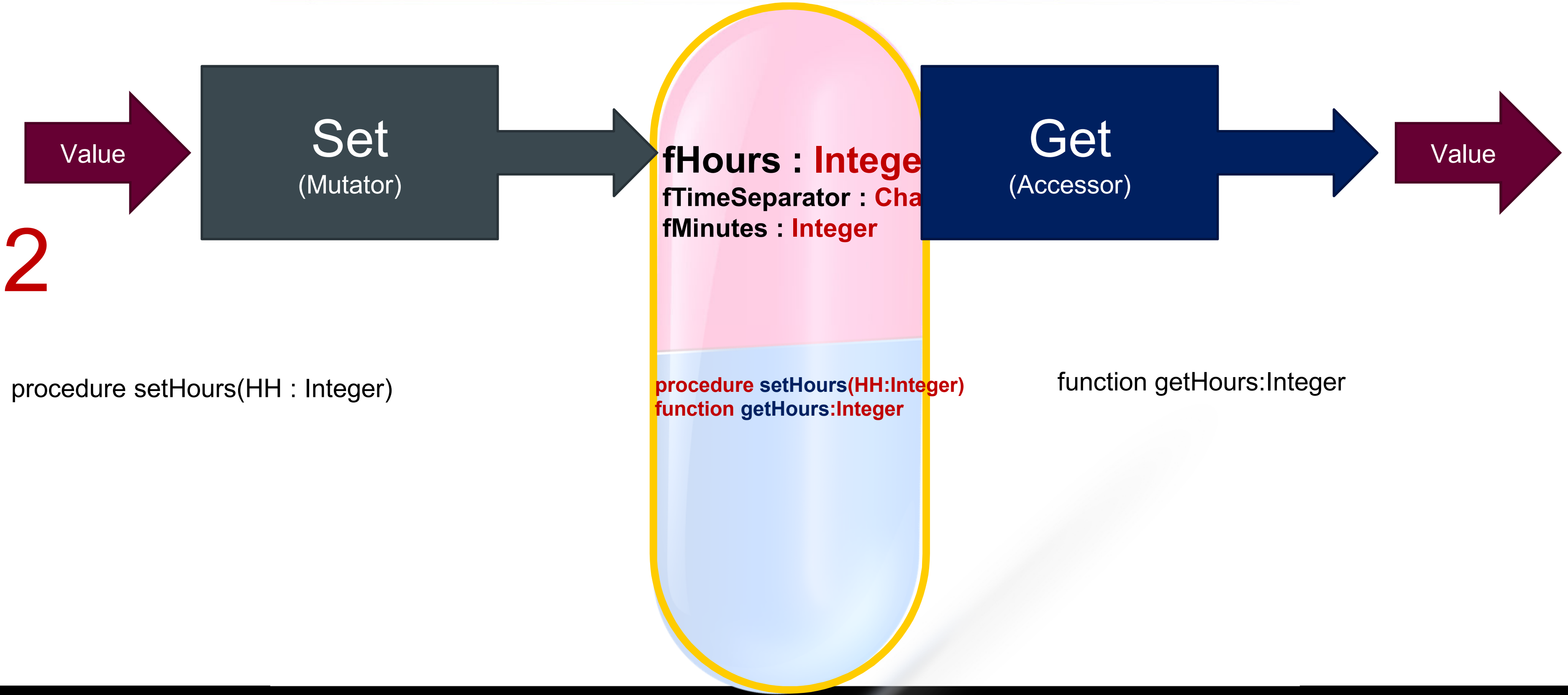
fHours : Integer
fTimeSeparator : Char
fMinutes : Integer

Step by Step Example

Class / ADT



Object



Value

Set
(Mutator)

fHours : Intege
fTimeSeparator : Cha
fMinutes : Integer

procedure setHours(HH:Integer)
function getHours:Integer

Get
(Accessor)

Value

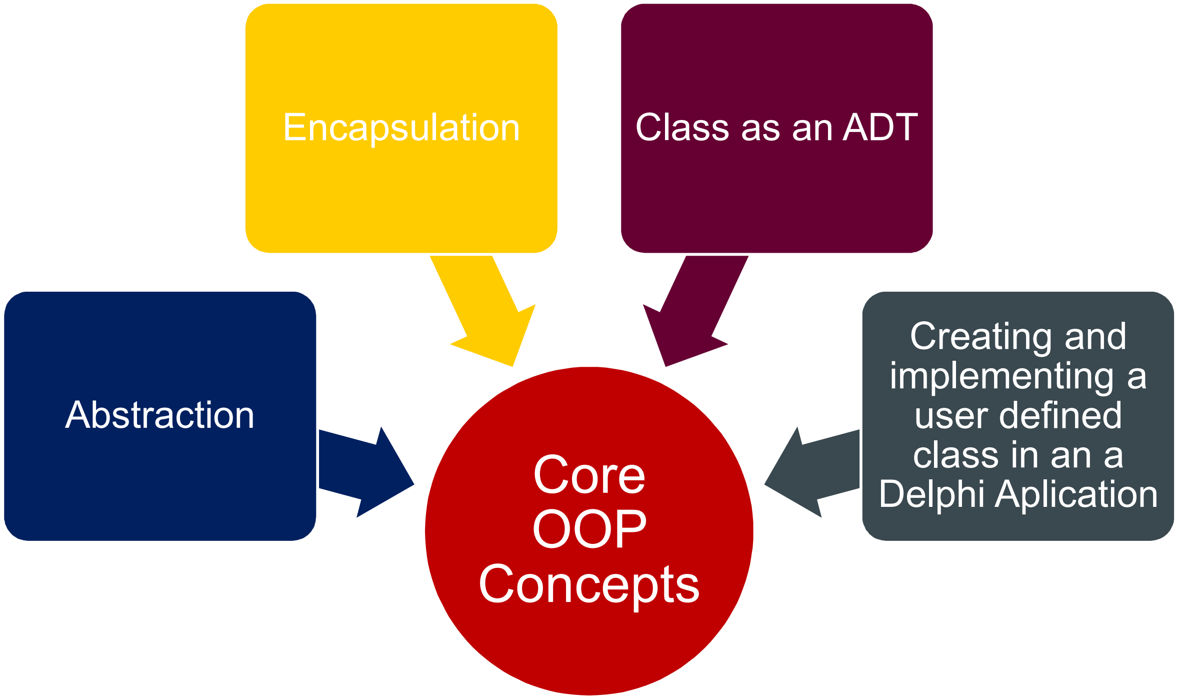
2

2

procedure setHours(HH : Integer)

function getHours:Integer

What we have covered



Encapsulation

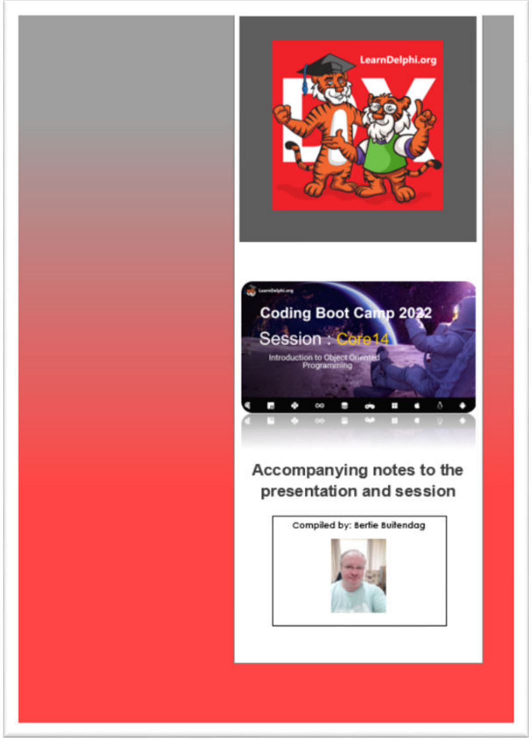
Class as an ADT

Abstraction

Creating and implementing a user defined class in an a Delphi Aplication

Core OOP Concepts

Additional notes + Example 2



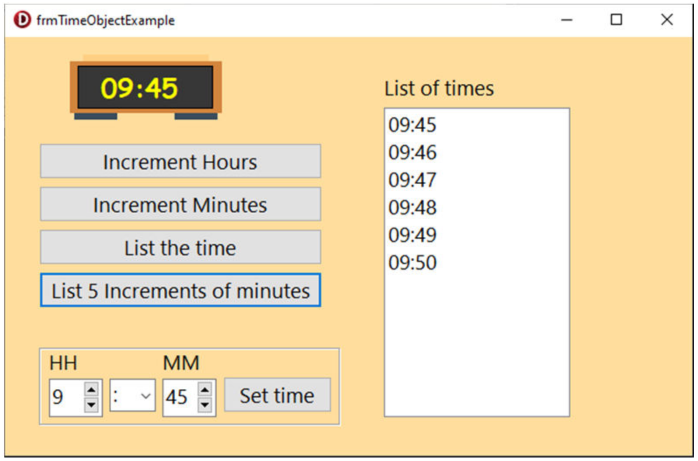
Coding Boot Camp 2022

Session : Core14

Introduction to Object Oriented Programming

Accompanying notes to the presentation and session

Compiled by: Bertie Buitendag



09:45

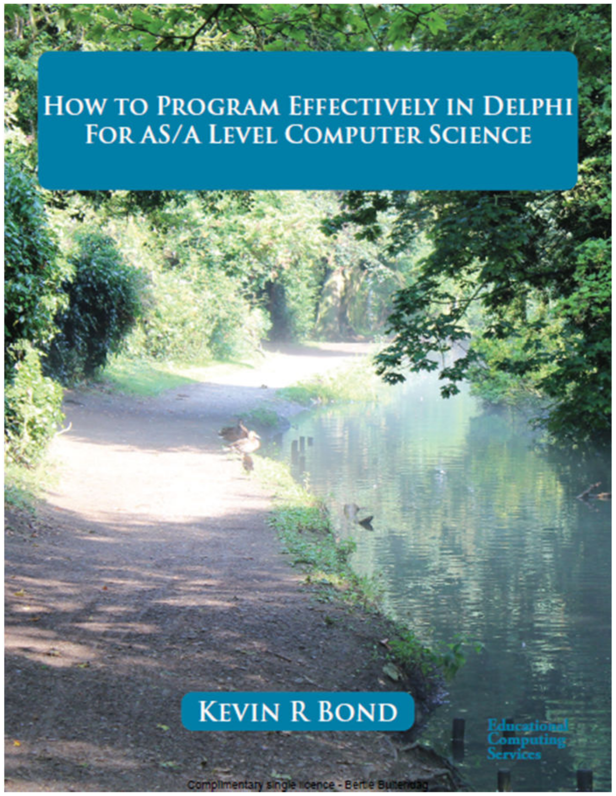
List of times

```
09:45
09:46
09:47
09:48
09:49
09:50
```

HH 9 : MM 45

Slides + extra examples and notes available on GitHub

Resources



HOW TO PROGRAM EFFECTIVELY IN DELPHI
FOR AS/A LEVEL COMPUTER SCIENCE

KEVIN R BOND

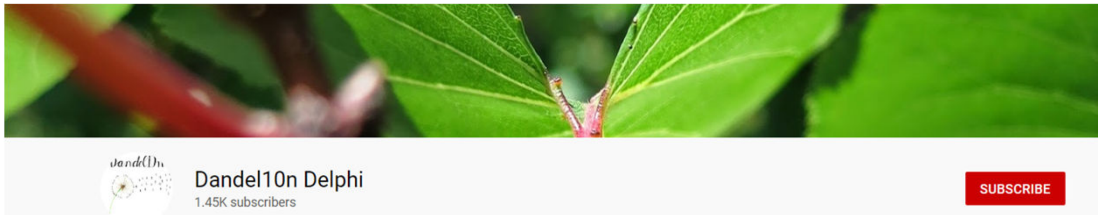
Educational
Computing
Services

Resources

WITH
Gerhard
LeARN
DeLPHI
MORE THAN 200 VIDEO TUTORIALS
VISUAL, ANIMATED EXPLANATIONS
STEP-BY-STEP, LINE-BY-LINE CODE WALKTHROUGHS
JOIN/SUPPORT US ON
Patreon.com/LearnDelphi

https://www.youtube.com/c/DelphiSchoolsDelphiProgrammingTutorials

https://www.youtube.com/channel/UCPQG0dcSGvOP20NLcjnYTPQ



https://www.youtube.com/channel/UCITMwGlsiKgY5Z-F5ZaCH4w/videos

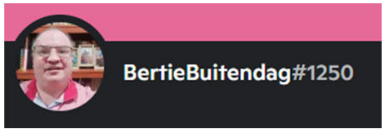
Resources

Alister Christie
8.86K subscribers
SUBSCRIBE

```
Label1.Caption := ‘Thank you 😊 ’;
```

Questions and Answers posts

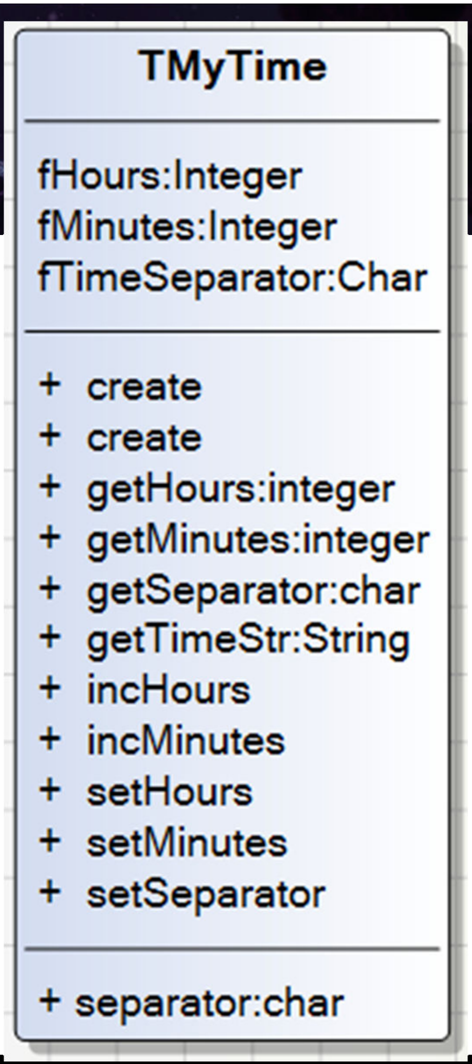
Discord



BertieBuitendag#1250

Label1.Caption := ‘Thank you 😊 ’;

Class diagram



TMyTime

fHours:Integer
fMinutes:Integer
fTimeSeparator:Char

+ create
+ create
+ getHours:integer
+ getMinutes:integer
+ getSeparator:char
+ getTimeStr:String
+ incHours
+ incMinutes
+ setHours
+ setMinutes
+ setSeparator

+ separator:char

LearnDelphi.org

Thank you!

Get the code:
github.com/Embarcadero/CodingBootCamp2022